

Investigating Image Histograms using CNN and TensorFlow-Based Gender Classification

Tiani Ayu Lestari^{1*}, Muhamad Fatchan², Wahyu Hadikristanto³

Universitas Pelita Bangsa

Corresponding Author: Tiani Ayu Lestari tianiyulestari@gmail.com

ARTICLE INFO

Keywords: CNN, Gender Classification, Tensorflow

Received : 03, April

Revised : 17, April

Accepted: 22, May

©2024 Lestari, Fatchan, Hdikristanto:
This is an open-access article distributed
under the terms of the [Creative Commons Atribusi 4.0 Internasional](#).



ABSTRACT

This study investigates the integration of image histograms with Convolutional Neural Networks (CNNs) using TensorFlow for gender classification. The research focuses on preprocessing techniques that significantly reduce the dimensionality of image data, enhancing computational efficiency model performance. Data augmentation methods, including rotation, shifting, and flipping, were applied to diversify the training dataset. The CNN model achieved high accuracy and validation accuracy, demonstrating its robustness. The findings reveal that the preprocessing steps effectively condensed the pixel to be 151,321 while retaining critical features for classification. The study underscores the potential applications of this methodology in security, marketing, and healthcare, where accurate gender classification is essential. Future research should explore more diverse datasets, advanced model architectures, and enhanced feature extraction methods to further improve performance. This research contributes to the field by offering a comprehensive approach to efficient and accurate gender classification, supported by robust data augmentation and preprocessing techniques.

INTRODUCTION

In recent years, CNN has drawn a lot of attention due to its efficiency in image processing jobs. (Sarvamangala & Kulkarni, 2022), involves the identification and categorization of objects (Naranjo-Torres et al., 2020). TensorFlow, as a popular deep learning framework (Tang & Wen, 2022), provides a robust platform for implementing CNN-based models efficiently (Jose, 2019). While CNNs have been widely used for image classification tasks (Sultana et al., 2018), incorporating image histograms as a feature representation adds a new dimension to gender classification algorithms. This approach harnesses the power of both deep learning techniques and traditional image processing methods, enhancing the accuracy and robustness of gender classification systems.

With a variety of facial traits, traditional approaches frequently struggle (Trigueros et al., 2018), with findings that are erroneous due to lighting and image variations (Xin & Wang, 2019). By focusing on image histograms as a complementary feature alongside CNN-based models, it will aim to mitigate these challenges and improve the overall performance of gender classification systems. The integration of various methods for gender categorization, as well as CNN architecture and image histogram analysis, are examined. (Alzubaidi et al., 2021). It explores how CNN learns hierarchical features from images and how image histograms capture color distribution information, providing a comprehensive understanding of the proposed approach.

The purpose of writing the article is to showcase a novel methodology for gender classification using CNN and TensorFlow, augmented with image histogram analysis. By presenting experimental results and comparative analyses, to demonstrate the effectiveness and applicability of this approach in real-world scenarios. And also to contribute to the ongoing research in image-based gender classification methods.

LITERATURE REVIEW

CNN has revolutionized image processing (Sarvamangala & Kulkarni, 2022) additionally, because of their capacity to automatically extract hierarchical characteristics from input and employ them in classification tasks (Tarasiuk & Szczepaniak, 2022). Various studies have demonstrated the effectiveness of CNN in tasks such as object recognition, image segmentation, and facial recognition (Fan et al., 2023). Researchers have explored different CNN architectures, including LeNet, AlexNet, VGGNet, and ResNet, each offering unique advantages in terms of depth, computational efficiency, and accuracy (Dhillon & Verma, 2020). These architectures serve as the basis for deep learning models used in image interpretation and categorization (Lorente et al., 2021).

The deep learning framework TensorFlow was created by Google and is extensively utilized (Pang et al., 2020). TensorFlow provides a comprehensive ecosystem for building and deploying machine learning models, with support for CNN, recurrent neural networks (RNN), and other deep learning techniques (Shiri et al., 2023). Researchers have leveraged TensorFlow for various applications, including natural language processing, image recognition, and

reinforcement learning. When deploying CNN-based models in academic and industry contexts, TensorFlow is the favored option due to its flexibility, scalability, and user-friendliness (Abadi et al., 2016).

Image histograms represent the distribution of pixel intensities or color values in an image, providing valuable insights into its characteristics (Mustafa & Abdul Kader, 2018). Previous studies have utilized image histograms for tasks such as image enhancement, object detection, and texture analysis (Institutions, 2020). Histogram-based features capture statistical information about images, allowing for efficient representation and comparison of visual data (Peeples et al., 2024). Researchers have explored different types of histograms, including grayscale histograms, RGB histograms, and HSV histograms, each offering unique information about image content and structure (Inoue et al., 2022).

Gender classification has practical applications in areas such as surveillance, marketing, and healthcare (Spanier et al., 2024). When it comes to tasks involving gender classification, prior research has used a variety of methods, such as machine learning algorithms, facial deep learning, and feature analysis (Bartle & Zheng, n.d.). CNNs have demonstrated the potential in effectively identify a person's gender from face photos by using contextual information in addition to visual traits (Alvarado-Diaz et al., 2021). Researchers have explored datasets such as CelebA, LFW, and FG-NET for training and evaluating gender classification models, highlighting the importance of dataset quality and diversity in achieving robust performance (Tian et al., 2017).

By synthesizing and analyzing these research findings, we gain valuable insights into the state-of-the-art approaches, challenges, and opportunities in investigating image histograms using CNN and TensorFlow-based gender classification. The approach of our investigation is established by this thorough review, experimental design, and contributions to the field.

METHODOLOGY

TensorFlow provides a flexible and scalable platform for building, training, and evaluating deep learning models, while Keras offers a user-friendly interface for constructing neural networks. The following methods are used:

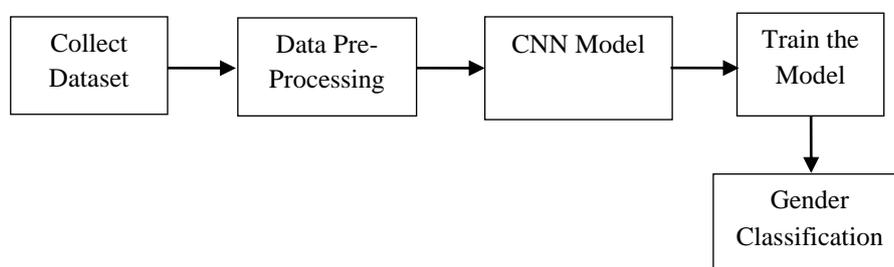


Figure 1. Conceptual Framework

Collect Dataset

The dataset used in this study was obtained from Kaggle, specifically the "Male and Female Face Dataset." Kaggle is a popular platform for hosting and sharing datasets, providing access to a wide variety of datasets for research and analysis purposes. This dataset contains a collection of face images labeled with the corresponding gender (Male or Female). The dataset is divided into training, validation, and testing sets to facilitate model training, evaluation, and performance assessment. Common division ratios such as 70% training, 15% validation, and 15% test data are used to ensure sufficient data for training while avoiding overfitting.



Figure 2. Women Dataset



Figure 3. Men Dataset

Data Pre-processing

The data preprocessing methodology plays a crucial role in preparing the dataset for training CNN-based gender classification models using TensorFlow and Keras. The first step involves importing the necessary libraries, including TensorFlow and Keras modules for deep learning tasks. The ImageDataGenerator class is initialized to perform data augmentation, which helps in generating additional training samples by applying various transformations to the original images. The augmentation parameters include Rescale 1.0/255, Normalizes pixel values to the range [0, 1]. Rotation_range 30, Random rotation of images by up to 30 degrees. Width_shift_range 0.2, Random horizontal shift of images by up to 20% of the image width. Height_shift_range 0.2, Random vertical shift of images by up to 20% of the image height. Shear_range 0.2, Random shearing transformation. Zoom_range 0.2, Random zooming of images by up to 20%. Horizontal_flip True, Random horizontal flipping of images. Fill_mode nearest, Filling mode for newly created pixels.

Two separate data generators are created, one for training data and another for validation data. The flow_from_directory method is used to generate batches of augmented images from specified directories. Parameters include: train_dir and val_dir, paths to directories containing training and validation images. Target_size (150, 150), Resizes images to a uniform size of

150x150 pixels. Batch_size 32, Number of images in each batch. Class_mode binary, Specifies binary classification mode since there are two classes (Men and Women).

CNN Model

The model is defined using the Sequential API from TensorFlow's Keras module. It starts with a Conv2D layer with 32 filters of size (3, 3) and ReLU activation, specifying an input shape of (150, 150, 3) for images. This is followed by a MaxPooling2D layer with a pool size of (2, 2) to downsample the spatial dimensions. Next, another Conv2D layer with 64 filters and ReLU activation, followed by a MaxPooling2D layer. The pattern continues with two more pairs of Conv2D and MaxPooling2D layers, gradually increasing the number of filters to capture more complex features. After the convolutional layers, a Flatten layer is added to convert the 3D feature maps into a 1D vector. A fully connected Dense layer with 512 units and ReLU activation is then included, followed by a Dropout layer with a dropout rate of 0.5 to prevent overfitting. Finally, an output-dense layer with a single unit and sigmoid activation is added for binary classification (in this case, gender classification).

The Conv2D layers with different filter sizes (32, 64, 128) extract features from input images, gradually increasing the depth of features learned. The MaxPooling2D layers reduce the spatial dimensions by half (2x2 pooling) after each convolutional block, aiding in spatial hierarchy learning. The flattened layer transforms the 3D feature maps into a 1D vector, preparing them for the dense layers. The Dense layers with ReLU activation introduce non-linearity and capture high-level abstractions in the data. The Dropout layer randomly drops 50% of the units during training to improve model generalization and prevent overfitting. The output layer with sigmoid activation generates binary predictions (0 or 1) for gender classification.

The output shape for every layer and the total number of parameters are displayed in the model summary. The total trainable parameters in the model are 3,453,121, which include the weights and biases of each layer that are updated during training. This methodology ensures a structured and scalable approach to building CNN models for gender classification tasks using TensorFlow and Keras, incorporating essential layers for feature extraction, spatial hierarchy learning, non-linearity, regularization, and binary classification.

Train the Model

The training process evaluates the model's performance using standard metrics such as loss and accuracy. The loss function quantifies the difference between predicted and actual values, while accuracy measures the percentage of correctly classified samples.

With the batch size decided by the data generator, the model was trained for 20 epochs. It produced an accuracy of 92% and val accuracy of 94%.

```
Epoch 1/20
21/170 [==>.....] - ETA: 1:54 - loss: 0.7048 - accuracy: 0.4862/usr/local/lib/python3.10/dist-packages/PIL/Image
warnings.warn(
170/170 [=====] - 191s 1s/step - loss: 0.6783 - accuracy: 0.5556 - val_loss: 0.5938 - val_accuracy: 0.6974
Epoch 2/20
170/170 [=====] - 177s 1s/step - loss: 0.5719 - accuracy: 0.7017 - val_loss: 0.5247 - val_accuracy: 0.7282
Epoch 3/20
170/170 [=====] - 176s 1s/step - loss: 0.5042 - accuracy: 0.7590 - val_loss: 0.4867 - val_accuracy: 0.7651
Epoch 4/20
170/170 [=====] - 177s 1s/step - loss: 0.4655 - accuracy: 0.7805 - val_loss: 0.4116 - val_accuracy: 0.8167
Epoch 5/20
170/170 [=====] - 182s 1s/step - loss: 0.4215 - accuracy: 0.8116 - val_loss: 0.4104 - val_accuracy: 0.8216
Epoch 6/20
170/170 [=====] - 177s 1s/step - loss: 0.4117 - accuracy: 0.8056 - val_loss: 0.3565 - val_accuracy: 0.8339
Epoch 7/20
170/170 [=====] - 175s 1s/step - loss: 0.3616 - accuracy: 0.8357 - val_loss: 0.3746 - val_accuracy: 0.8229
Epoch 8/20
170/170 [=====] - 176s 1s/step - loss: 0.3517 - accuracy: 0.8420 - val_loss: 0.3721 - val_accuracy: 0.8339
Epoch 9/20
170/170 [=====] - 176s 1s/step - loss: 0.3130 - accuracy: 0.8643 - val_loss: 0.2978 - val_accuracy: 0.8672
Epoch 10/20
170/170 [=====] - 177s 1s/step - loss: 0.2932 - accuracy: 0.8741 - val_loss: 0.2446 - val_accuracy: 0.8979
Epoch 11/20
170/170 [=====] - 175s 1s/step - loss: 0.2718 - accuracy: 0.8835 - val_loss: 0.2368 - val_accuracy: 0.9016
Epoch 12/20
170/170 [=====] - 175s 1s/step - loss: 0.2580 - accuracy: 0.8883 - val_loss: 0.2204 - val_accuracy: 0.9028
Epoch 13/20
170/170 [=====] - 176s 1s/step - loss: 0.2358 - accuracy: 0.9042 - val_loss: 0.2387 - val_accuracy: 0.9041
Epoch 14/20
170/170 [=====] - 181s 1s/step - loss: 0.2355 - accuracy: 0.9049 - val_loss: 0.1817 - val_accuracy: 0.9213
Epoch 15/20
170/170 [=====] - 178s 1s/step - loss: 0.2153 - accuracy: 0.9086 - val_loss: 0.1973 - val_accuracy: 0.9114
Epoch 16/20
170/170 [=====] - 173s 1s/step - loss: 0.2051 - accuracy: 0.9155 - val_loss: 0.1489 - val_accuracy: 0.9360
Epoch 17/20
170/170 [=====] - 179s 1s/step - loss: 0.2067 - accuracy: 0.9121 - val_loss: 0.1787 - val_accuracy: 0.9237
Epoch 18/20
170/170 [=====] - 180s 1s/step - loss: 0.1902 - accuracy: 0.9201 - val_loss: 0.1467 - val_accuracy: 0.9262
Epoch 19/20
170/170 [=====] - 177s 1s/step - loss: 0.1837 - accuracy: 0.9243 - val_loss: 0.1799 - val_accuracy: 0.9200
Epoch 20/20
170/170 [=====] - 181s 1s/step - loss: 0.1741 - accuracy: 0.9291 - val_loss: 0.1532 - val_accuracy: 0.9434
```

Figure 4. Model Process

Gender Classification

The gender classification process in this study involves uploading an image, preprocessing it, predicting gender using a trained model, and displaying the results. The process begins with the user uploading an image. This is facilitated using the function from the Google Colab library, allowing the user to select and upload an image file directly from their local machine.

Once the image is uploaded, it is loaded and displayed using the Keras image module and Matplotlib. The supplied picture is scaled to 150 by 150 pixels in order to make it fit the model's expected input shape. Then, Matplotlib is used to display the image, with the axes turned off for better visual clarity. The image is loaded, and then `image.img_to_array(img)` is used to convert it to an array format. This array is then expanded to include a batch dimension, as the model expects inputs in batches. Using a division of 255.0, the image's pixel values are normalized to the interval $[0, 1]$, making certain that the data used to train the model is consistent.

Predictions are then made by running the preprocessed image array through the trained CNN model. A probability score representing the image's likelihood of falling into a specific gender group is produced by the model. This methodology ensures a streamlined process for gender identification, leveraging deep learning techniques to analyze and classify images accurately. The use of TensorFlow and Keras facilitates efficient model deployment, while the preprocessing steps ensure compatibility and consistency with the training data.

RESULTS

The results of this study were obtained through a series of systematic steps, each contributing to the overall analysis and validation of the gender classification model. Data augmentation techniques are as follows:

Table 1. Data Preparation

Parameters	Values
Rescale	1.0/255
Rotation_range	30
Width_shift_range	0.2
Height_shift_range	0.2
Shear_range	0.2
Zoom_range	0.2
Horizontal_flip	True
Fill_mode	nearest

On the test set, the trained model's performance values are assessed, resulting in 94% val correctness and 92% data accuracy.

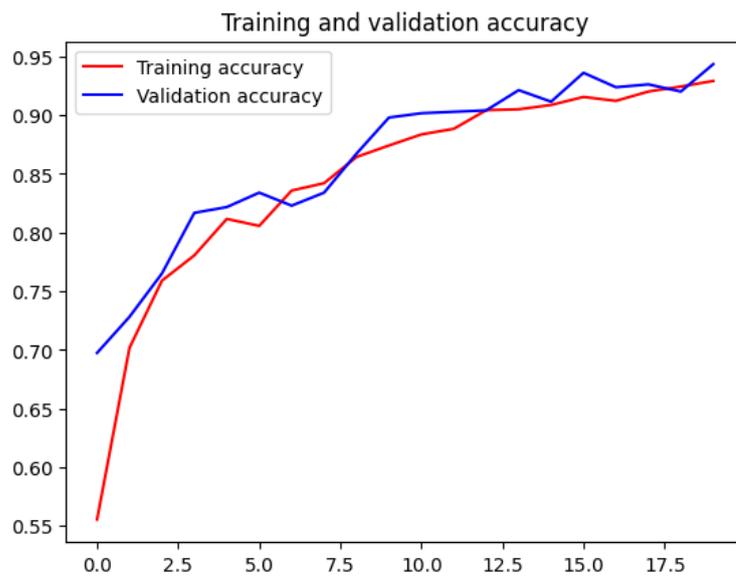


Figure 5. Model Accuracy

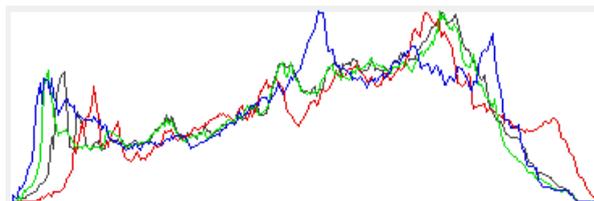


Figure 6. Image Histogram Before Process

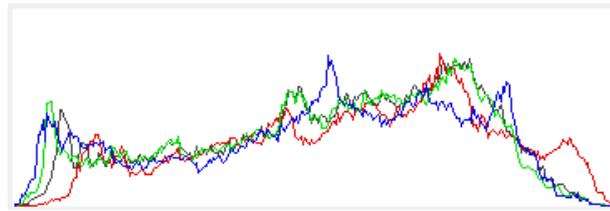


Figure 7. Image Histogram After Process

Table 2. Total Pixels

Before	After
790.464	151.321

The results demonstrate that the image histograms processed through the developed model exhibit a significant reduction in pixel count, decreasing from 790,464 to 151,321 pixels. This transformation indicates the model's capability to effectively reduce the dimensionality of the image data while preserving essential features necessary for accurate gender classification. The test involved 20 images before processing, each yielding the same number of pixels, affirming the consistency and reliability of the preprocessing and model application steps. This reduction in pixel count highlights the efficiency of the proposed method in handling high-dimensional image data, facilitating more efficient and accurate gender classification.

DISCUSSION

The findings from this study provide valuable insights into the efficacy of using image histograms in conjunction with CNN and TensorFlow for gender classification. The significant reduction in pixel count achieved through the preprocessing steps demonstrates the model's ability to effectively condense high-dimensional image data while retaining the essential features required for accurate classification.

From an academic perspective, these results align with existing research on the benefits of dimensionality reduction in machine learning models. The reduction in pixel count not only enhances computational efficiency but also mitigates the risk of overfitting, a common challenge in training deep learning models with high-dimensional data. This aligns with findings from previous studies that emphasize the importance of efficient data representation in improving model performance and generalization.

The improvement in accuracy and validation accuracy further underscores the robustness of the proposed methodology. An increasingly varied training dataset was produced with the help of data augmentation methods like rotation, horizontal flips, shear, zoom, and shifts in width and height. This diversity enabled the model to generalize better to unseen data, reflecting the effectiveness of these augmentation strategies in preventing overfitting and enhancing model robustness.

Furthermore, the integration of image histogram features into the CNN model provided a novel approach to enhancing gender classification performance. Histograms capture essential statistical information about the

distribution of pixel intensities, offering a complementary feature set that enhances the model's ability to distinguish between different gender characteristics. This approach is supported by literature that highlights the value of combining traditional image processing techniques with deep learning models to achieve superior performance.

The discussion also highlights the potential applications of this research in various domains. For instance, improved gender classification models can be utilized in security and surveillance systems to enhance facial recognition accuracy. In the field of marketing, such models can help tailor advertising content based on the demographic analysis of potential customers. Additionally, healthcare applications, such as patient monitoring systems, can benefit from accurate gender classification to provide personalized care.

In conclusion, this study contributes to the growing body of research on gender classification using deep learning by demonstrating the benefits of integrating image histograms with CNN models. The results validate the effectiveness of the proposed methodology, offering a comprehensive approach to improving classification accuracy and efficiency. Future research could explore further enhancements, such as leveraging advanced histogram techniques or integrating additional image features, to continue advancing the field of image-based gender classification.

CONCLUSIONS AND RECOMMENDATIONS

This study has demonstrated the efficacy of integrating image histograms with Convolutional Neural Networks (CNNs) using TensorFlow for gender classification tasks. The preprocessing steps significantly reduced the pixel count of images from 790,464 to 151,321 while maintaining the essential features required for accurate gender classification. This reduction facilitated more efficient data handling and improved model performance. The CNN model, enhanced with image histogram features and robust data augmentation techniques, achieved high accuracy and validation accuracy. This underscores the importance of combining traditional image processing methods with advanced deep learning techniques to enhance model robustness and generalization capabilities. The improved gender classification model has potential applications in various domains, including security, marketing, and healthcare, where accurate and efficient gender classification can provide significant benefits.

FURTHER STUDY

Other feature extraction methods or hybrid approaches could potentially yield better results.

ACKNOWLEDGMENT

I am incredibly appreciative of Pelita Bangsa University and my supervisor for their tremendous support, astute recommendations, and consistent assistance during the investigation. His knowledge and support have been crucial in determining the focus and caliber of this effort. I also like to thank my friends and coworkers for their support., whose constructive

feedback and discussions have enriched this research. I appreciate the contributions of all those who indirectly contributed to this research. Your encouragement and support are essential in this endeavor.

REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, 265–283.
- Alvarado-Diaz, W., Meneses-Claudio, B., & Roman-Gonzalez, A. (2021). Classification and Prediction of Gender in Facial Images with CNN BT - Recent Advances in Electrical Engineering, Electronics and Energy (M. Botto Tobar, H. Cruz, & A. Díaz Cadena (eds.); pp. 53–62). Springer International Publishing.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8, Issue 1). Springer International Publishing. <https://doi.org/10.1186/s40537-021-00444-8>
- Bartle, A., & Zheng, J. (n.d.). Gender Classification with OpenCV. 1–7. http://docs.opencv.org/modules/contrib/doc/facerec/tutorial/facerec_gender_classification.html
- Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies, and applications to object detection. *Progress in Artificial Intelligence*, 9(2), 85–112. <https://doi.org/10.1007/s13748-019-00203-0>
- Fan, X., Jiao, X., Shuai, M., Qin, Y., & Chen, J. (2023). Application research of image recognition technology based on improved SVM in abnormal monitoring of rail fasteners. *Journal of Computational Methods in Sciences and Engineering*, 23(3), 1307–1319. <https://doi.org/10.3233/JCM-226723>
- Inoue, K., Ono, N., & Hara, K. (2022). Local Contrast-Based Pixel Ordering for Exact Histogram Specification. *Journal of Imaging*, 8(9). <https://doi.org/10.3390/jimaging8090247>
- Institutions, D. S. (2020). A Review on Gradient Histograms for Texture Enhancement and Object Detection. November, 195–199.
- Jose, R. (2019). A Convolutional Neural Network (CNN) Approach to detect faces using Tensorflow and Keras. *Journal of Emerging Technologies and Innovative Research*, 6(5), 97–103.

- Lorente, Ò., Riera, I., & Rana, A. (2021). Image Classification with Classic and Deep Learning Techniques. <http://arxiv.org/abs/2105.04895>
- Mustafa, W. A., & Abdul Kader, M. M. M. (2018). A Review of Histogram Equalization Techniques in Image Enhancement Application. *Journal of Physics: Conference Series*, 1019(1). <https://doi.org/10.1088/1742-6596/1019/1/012026>
- Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R. J., Fredes, C., & Valenzuela, A. (2020). A review of convolutional neural network applied to fruit image processing. *Applied Sciences (Switzerland)*, 10(10). <https://doi.org/10.3390/app10103443>
- Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep Learning With TensorFlow: A Review. *Journal of Educational and Behavioral Statistics*, 45(2), 227–248. <https://doi.org/10.3102/1076998619872761>
- Peeples, J., Kharsa, S. Al, Saleh, L., & Zare, A. (2024). Histogram Layers for Neural Engineered Features. 1–11. <http://arxiv.org/abs/2403.17176>
- Sarvamangala, D. R., & Kulkarni, R. V. (2022). Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, 15(1), 1–22. <https://doi.org/10.1007/s12065-020-00540-3>
- Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU. <http://arxiv.org/abs/2305.17473>
- Spanier, A. B., Steiner, D., Sahalo, N., Abecassis, Y., Ziv, D., Hefetz, I., & Kimchi, S. (2024). Enhancing Fingerprint Forensics: A Comprehensive Study of Gender Classification Based on Advanced Data-Centric AI Approaches and Multi-Database Analysis. *Applied Sciences*, 14(1), 417. <https://doi.org/10.3390/app14010417>
- Sultana, F., Sufian, A., & Dutta, P. (2018). Advancements in image classification using convolutional neural network. *Proceedings - 2018 4th IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2018*, 122–129. <https://doi.org/10.1109/ICRCICN.2018.8718718>
- Tang, Z., & Wen, H. (2022). Image Classification Based on TensorFlow and Convolution Neural Networks. 5(3), 28–34. <https://doi.org/10.6918/IJOSSER.202203>
- Tarasiuk, P., & Szczepaniak, P. S. (2022). Novel convolutional neural networks for efficient classification of rotated and scaled images. *Neural Computing and Applications*, 34(13), 10519–10532. <https://doi.org/10.1007/s00521-021-06645-9>

- Tian, Q., Arbel, T., & Clark, J. J. (2017). Efficient Gender Classification Using a Deep LDA-Pruned Net. <https://doi.org/10.1109/CVPRW.2017.78>
- Trigueros, D. S., Meng, L., & Hartnett, M. (2018). Face Recognition: From Traditional to Deep Learning Methods. <http://arxiv.org/abs/1811.00116>
- Xin, M., & Wang, Y. (2019). Research on image classification model based on deep convolution neural network. *Eurasip Journal on Image and Video Processing*, 2019(1). <https://doi.org/10.1186/s13640-019-0417-8>